

قد تحتاج كثير من التطبيقات العملية الى ربط المتحكم (مثل البيك) مع عناصر خارجية لاداء الوظيفة المطلوبة .. وكما كان عدد الاطراف اللازمة للتوصيل اقل ما يمكن كلما كان هناك توفير في عدد اطراف المتحكم وبالتالي تستخدم لاداء الوظائف الاخرى للمشروع المطلوب .. حيث تعتمد هذه الطريقة على نقل الداتا بشكل تسلسلي من خلال الطرف المحدد لذلك.

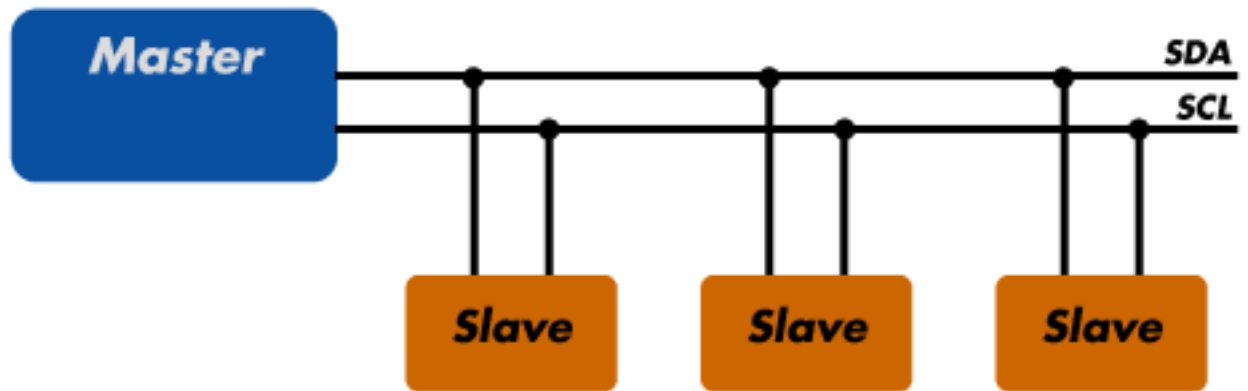
من هنا كان اختيار بروتوكول الربط I2C من اكثر الطرق المستخدمة للربط بين المتحكمات وبين العناصر الالكترونية المختلفة مثل الذاكرة الخارجية التسلسلية (EEPROM).

تتألف طريقة الربط بواسطة I2C من طرفين للتوصيل :

١ - طرف لتوصيل نبضات الساعة SCL.

٢ - طرف لتوصيل الداتا SDA.

يمكن من خلال بروتوكول I2C توصيل عناصر متعددة الى نفس الطرفين والتخاطب معها جميعا بدون ان يحدث تداخل بينها .



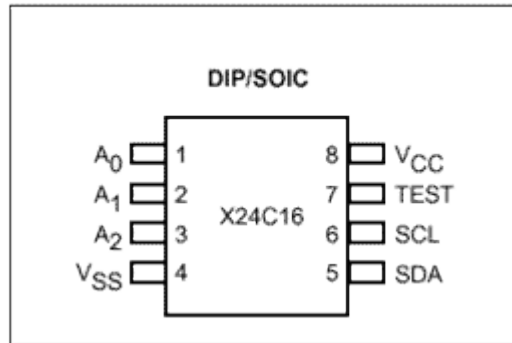
والسؤال الذي يطرح نفسه : في حال كان لدي عدة عناصر الكترونية (مثلا لدي ذاكرة EEPROM وقطعه اخرى ولتكن DS1307) والتي تستخدم طريقة I2C فكيف يمكن نقل الداتا من المتحكم الى الايبروم فقط ؟؟؟

الجواب بسيط : تعتمد عملية نقل الداتا على استخدام عنوان مميز لكل عنصر (هذا العنوان يشبه الاسم والذي يميز كل انسان) فمثلا الذاكرة الخارجية التسلسلية EEPROM لها رقم مميز يختلف عن DS1307 ، ويجب مراجعة الداتاشيت للعنصر لمعرفة هذا الرقم حتى نتمكن من نقل الداتا والا فان عملية الربط ستكون بدون جدوى .

تعتمد عملية نقل الداتا من خلال بروتوكول I2C على وجود طرفين يتم نقل الداتا بينهما :

- ١ - السيد او الماستر : وهو الذي يقوم بالتحكم بعملية النقل بشكل عام وهو المسؤول عن توليد نبضات الساعة اللازمة لعملية نقل الداتا ، وخلال الشرح سيكون المتحكم هو الذي يقوم بهذه الوظيفة ، وسنستخدم في الشرح PIC16F877A والذي يحتوي على هاردوير داخلي I2C .
- ٢ - التابع او الخادم : وهو الذي يقوم بكل ما يطلب منه من خلال الماستر مثل حفظ المعلومات او استرجاعها ، وخلال الشرح ستكون ذاكرة الايبروم الخارجية هي التي تقوم بهذه الوظيفة ، وسنستخدم الذاكرة 24C16.

PIN CONFIGURATION



ملاحظة مهمة : في حال استخدام الذاكرة 24C16 فان الذاكرة الداخلية تحتوي على 16384 Bit وليس 16384 byte ، وبالتالي للحصول على عدد البايتات في الذاكرة يجب ان نقسم الحجم المذكور على 8 ، هذا الامر قد يسبب بعض الاربك احيانا .

ان عملية نقل الداتا من المتحكم الى الايبروم تسمى عملية الكتابة الى الايبروم ، اما عملية نقل الداتا من الايبروم الى المتحكم فتسمى عملية القراءة من الايبروم .

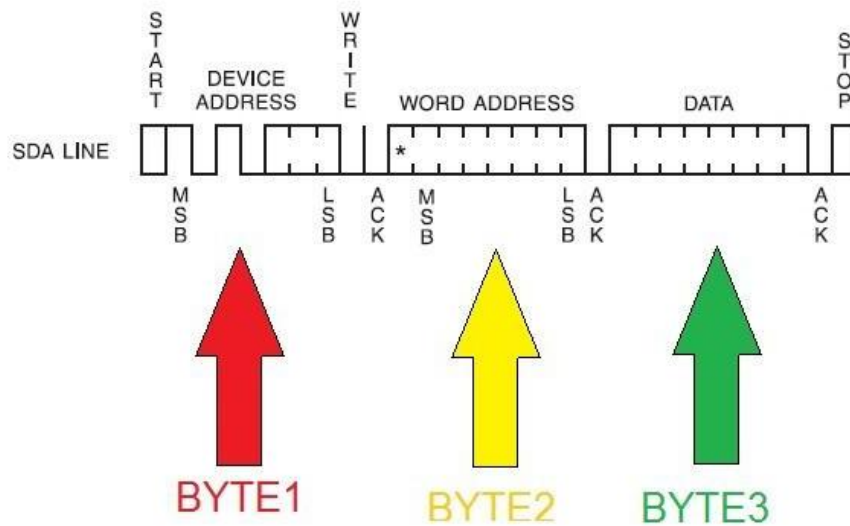
تتم عملية الكتابة الى الايبروم بطريقتين :

- ١ - كتابة بايت واحد فقط الى الايبروم (Byte Write) .
- ٢ - كتابة عدة بايتات متتالية الى الايبروم (Page Write) .

نبدأ مع النوع الاول :

- ١ - كتابة بايت واحد فقط الى الايبروم (Byte Write) .

لتوضيح عملية كتابة بايت الى الايبروم فلننظر الى الصورة التالية التي توضح تسلسل عملية الكتابة :



- ١ - تبدأ عملية ارسال الداتا بارسال اشارة بدء الارسال (START) من قبل الماستر .
- ٢ - يقوم الماستر بارسال البايت الاول BYTE1 والذي يتكون من 8 بتات لتحديد العنوان المميز للخادم ، في حال البت الثامن يساوي الصفر فان الخادم يفهم بان العملية هي عملية كتابة الى الايبروم ، اما في حال كان البت الثامن يساوي الواحد فان الخادم يفهم بان العملية هي عملية قراءة من الايبروم ، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال قد تمت بنجاح .
- ٣ - يقوم الماستر بارسال البايت الثاني BYTE2 والذي يحوي العنوان الذي سيتم الكتابة اليه داخل الايبروم ، ويمكن ان نصل الى 256 موقع ذاكرة (256 بايت) داخل الايبروم (هذا العنوان يتم وضعه في مؤشر الموقع للايبروم) ، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال قد تمت بنجاح .
- ٤ - واخيرا يقوم الماستر بارسال البايت الثالث BYTE3 ، والذي يحوي الداتا المطلوب كتابتها وحفظها في المكان الذي يدل عليه مؤشر الموقع للايبروم ، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال قد تمت بنجاح .
- ٥ - تنتهي عملية ارسال الداتا بارسال اشارة نهاية الارسال (STOP) من قبل الماستر .

ملاحظة مهمة في حال الايبروم المستخدم هنا :

من المهم لاتمام عملية الخاطب معرفة ان الايبروم تمتلك العنوان الثنائي المميز (1010) مهما كان حجم الايبروم ،
والسؤال الذي يطرح نفسه : ماذا نفعل بالبتات الثلاث الباقية في البايت الاول BYTE1 ؟؟؟

الجواب كالتالي:

قلنا ان الذاكرة تحتوي 16384 بت وهذا يعني 2048 بايت . فكم بت نحتاج لتمثيل هذه المواقع في الذاكرة ؟؟؟

بالحساب نجد ان : $2^{11}=1048$

اي اننا نحتاج الى Bit 11 حتى نصل الى اي موقع موجود في الذاكرة .. وبالتالي نستخدم البتات الثلاثة الموجودة في البايت الاول BYTE1 مع كامل البايت الثاني BYTE2 للحصول على Bit 11 وبالتالي الوصول الى اي موقع في الايبروم ، وهكذا يتم الوصول الى اي موقع داخل الايبروم من خلال مؤشر الموقع ،حيث يزداد المؤشر بعد كل عملية قراءة او كتابة بشكل تلقائي.

ما رأيكم الان ان ننقل الى الجزء العملي في المحاضرة

المثال الاول

و لنقم قم بعمل برنامج باستخدام المايكروسي يقوم بحفظ القيمة 0xAA في الموقع رقم 1000 داخل الايبروم 24C16 وذلك باستخدام بروتوكول النقل I2C .

في بداية اي برنامج يستخدم I2C يجب تحديد التردد الذي يستخدم لنقل الداتا ، عادة فان تردد 100 كيلو هيرتز مناسب لكثير من التطبيقات . الامر المستخدم لتحديد التردد حسب المايكروسي هو نفس أمر التهيئة لننا كما ذكرنا في المجاضرات السابقة ان كل أمر في المايكروسي له داله تقوم بعمل التهيئة له وهذه هي الداله لعم التهيئة للـ I2C

```
I2C1_Init(100000);
```

البايت الاول : لتحديد البايت الاول نحن بحاجة الى العنوان المميز للايبروم وهو (1010)، والعملية هي كتابة الى الايبروم فأخر بت هو صفر .. يبقى الان الثلاث بتات التي تحدثنا عنها حيث نأخذها من السؤال (الموقع هو 1000)

وبالتالي فالموقع بالبايناري هو 1000 1110 011 = 1000

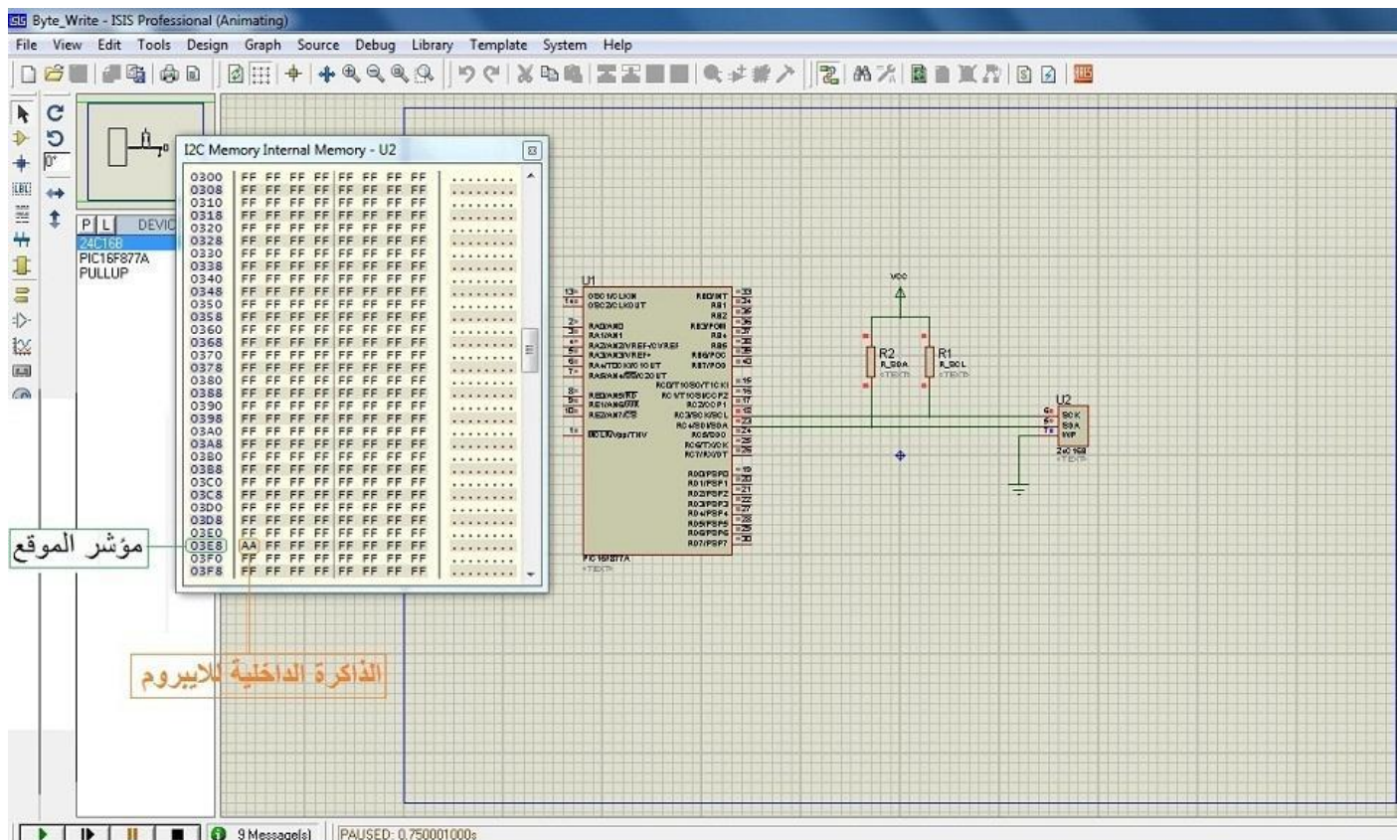
وهكذا فان البايت الاول هو :

Byte1=1010 0110

اما البايت الثاني فنأخذ بقية رقم الموقع وبالتالي يكون البرنامج كاملاً كما يلي

```
void main()
{
I2C1_Init(100000);
I2C1_Start();
I2C1_Wr(0xA6);
I2C1_Wr(0xE8);
I2C1_Wr(0xAA);
I2C1_Stop();
}
```

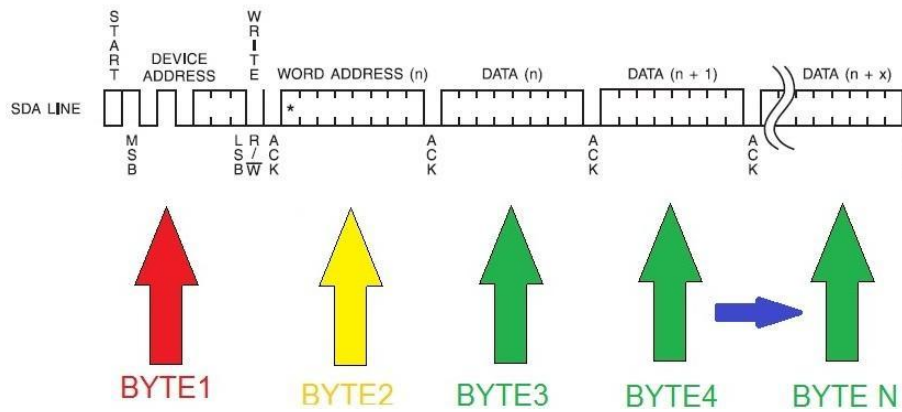
وبالنسبة لدائرة الكهربائية هي كما يلي مرسومه على برنامج البوروتس



نكمل مع النوع الثاني من طرق الكتابة وهي :

٢ - كتابة عدة بايتات متتالية الى الايبروم (Page Write) :

لا تختلف عملية كتابة بايت الى الايبروم عن كتابة عدة بايتات متتالية والصورة التالية توضح ذلك :



١ - تبدأ عملية ارسال الداتا بارسال اشارة بدء الارسال (START) من قبل الماستر .

٢ - يقوم الماستر بارسال البايت الاول BYTE1 والذي يتكون من 7 بتات لتحديد العنوان المميز للخدم ، ففي حال كان البت الثامن يساوي الصفر فان الخادم يفهم بان العملية هي عملية كتابة الى الايبروم ، اما في حال كان البت الثامن يساوي الواحد فان الخادم يفهم بان العملية هي عملية قراءة من الايبروم ، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال قد تمت بنجاح .

٣ - يقوم الماستر بارسال البايت الثاني BYTE2 والذي يحوي العنوان الذي سيتم الكتابة اليه داخل الايبروم ، ويمكن ان نصل الى 256 موقع ذاكرة (256 بايت) داخل الايبروم (هذا العنوان يتم وضعه في مؤشر الموقع للايبروم) ، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال قد تمت بنجاح .

٤ - الان يقوم الماستر بارسال البايت الثالث BYTE3 ، والذي يحوي الداتا المطلوب كتابتها وحفظها في المكان الذي يدل عليه مؤشر الموقع للايبروم ، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال قد تمت بنجاح .

ملاحظة مهمة : بعد عملية الكتابة هنا يزداد مؤشر الموقع بمقدار واحد تلقائيا .

٥ - بما ان مؤشر الموقع قد ازداد بمقدار 1 تلقائيا ، الان يقوم الماستر بارسال البايت الرابع BYTE4 كما ورد في النقطة السابقة ، وتعاد نفس العملية مادام هناك بايتات اخرى حتى البايت الاخير BYTE N .

٦ - تنتهي عملية ارسال الداتا بارسال اشارة نهاية الارسال (STOP) من قبل الماستر .

والان لنقم برنامج باستخدام الميكروسي يقوم بحفظ القيم (0xA0,0xA1,0xA2,0xA3,0xA4,0xA5) بداية من الموقع رقم 10 داخل الايبروم 24C16 وذلك باستخدام بروتوكول النقل I2C .

نقوم أولاً بتحديد البايتات اللازمة لعملية الكتابة الى الايبروم .

البايت الاول : لتحديد البايت الاول نحن بحاجة الى العنوان المميز للايبروم وهو (1010) ، والعملية هي كتابة الى الايبروم فأخر بت هو صفر ..يبقى الان الثلاث بتات التي تحدثنا عنها حيث نأخذها من السؤال (الموقع هو 10) :

الان الموقع بالبايناري هو 1010 0000 000 = 10

وهكذا فان البايت الاول هو : byte1 = 1010 000 0

البايت الثاني : هو البايت الذي يحدد باقي مكان مؤشر الموقع (الموقع هو 10) :

Byte2 = 0000 1010 = 0A

البايت الثالث : وهو عبارة عن الداتا التي سنحفظها في المكان الذي يدل عليه مؤشر الموقع (الموقع هو ١٠) :

Byte3 = 0xA0

البايت الرابع : وهو عبارة عن الداتا التي سنحفظها في المكان الذي يدل عليه مؤشر الموقع (الموقع هو ١١) :

Byte4 = 0xA1

البايت الخامس : وهو عبارة عن الداتا التي سنحفظها في المكان الذي يدل عليه مؤشر الموقع (الموقع هو ١٢) :

Byte5 = 0xA2

البايت السادس : وهو عبارة عن الداتا التي سنحفظها في المكان الذي يدل عليه مؤشر الموقع (الموقع هو ١٣) :

Byte6 = 0xA3

البايت السابع : وهو عبارة عن الداتا التي سنحفظها في المكان الذي يدل عليه مؤشر الموقع (الموقع هو ١٤) :

Byte7 = 0xA4

البايت الثامن : وهو عبارة عن الداتا التي سنحفظها في المكان الذي يدل عليه مؤشر الموقع (الموقع هو ١٥) :

Byte7 = 0xA5

وهكذا يكون البرنامج كاملا كالتالي:

```
void main() {
I2C1_Init(100000);
I2C1_Start();
I2C1_Wr(0xA0);
I2C1_Wr(0x0A);
I2C1_Wr(0xA0);
I2C1_Wr(0xA1);
I2C1_Wr(0xA2);
I2C1_Wr(0xA3);
I2C1_Wr(0xA4);
I2C1_Wr(0xA5);
I2C1_Stop();
}
```

الان سنوضح طريقة القراءة باستخدام هذا البروتوكول

ان تسلسل عملية القراءة لا تتم بنفس تسلسل عملية الكتابة ، وهذا يرجع لسبب رئيسي واحد : وهو انه لا يوجد طريقة للقراءة من عنوان محدد في الذاكرة ، بل نحن نحتاج ان نقوم بتحميل العنوان الذي سنقرأ منه اولاً (وهذا يتم باستخدام تسلسل عملية الكتابة المستخدمة في المشاركات السابقة) وبعدها نبدأ عملية القراءة .

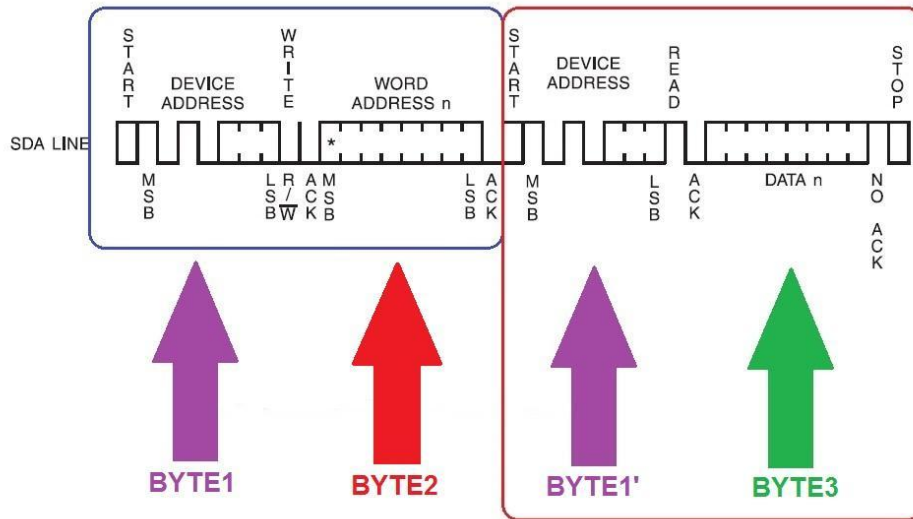
تتم عملية القراءة من الايبروم بطريقتين :

- ١ - قراءة بايت واحد فقط من الايبروم (Byte Read) .
- ٢ - قراءة عدة بايتات متتالية من الايبروم (Page Read) .

نبدأ مع النوع الاول وهو :

- ١ - كيفية قراءة بايت واحد من الايبروم (Byte Read) :

ان عملية القراءة من مكان محدد باستخدام هذا البروتوكول موضحة في الصورة التالية:



تتكون عملية القراءة من قسمين ، القسم الايسر المحدد باللون الازرق تم شرحه سابقا من خلال عملية الكتابة ، ووظيفته وضع مؤشر الموقع في المكان الذي سيتم القراءة منه كالتالي:

١ - ارسال اشارة بدء الارسال (START) من قبل الماستر .

٢ - يقوم الماستر بارسال البايت الاول BYTE1 والذي يتكون من ٧ بتات لتحديد العنوان المميز للخادم ، اما البت الثامن

فيساوي الصفر ، وهكذا يفهم الخادم بان العملية هي عملية كتابة الى الايبروم ، ، بعدها يرد الخادم باشارة ACK

والتي تعني بان عملية الاستقبال قد تمت بنجاح .

٣ - يقوم الماستر بارسال البايت الثاني BYTE2 والذي يحوي العنوان الذي سيتم الكتابة اليه داخل الايبروم ، هذا العنوان

يتم وضعه في مؤشر الموقع للايبروم (، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال قد تمت

بنجاح

نلاحظ ان هذه الخطوات هي نفس الخطوات المستخدمة للكتابة الى الايبروم ولكن مع الغاء اخر خطوة ، الان اصبح

مؤشر الموقع في المكان الذي نريد التعامل معه .

الان نبدا عملية القراءة من الموقع بعد ان قمنا بتحديد مكان مؤشر الموقع ، وهذا محدد في القسم الايمن المحدد باللون الاحمر كالتالي:

١ - يقوم الان الماستر بارسال اشارة بدء ارسال (START) مرة اخرى .

٢ - يقوم الماستر بارسال البايت ('BYTE1) وهو نفس البايت الاول (BYTE1) الذي يحدد العنوان المميز للخادم مع اختلاف واحد ، فالبت الثامن اصبح يحتوي القيمة (١) مما يعني ان العملية هي قراءة من الايبروم، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال للعنوان تمت بنجاح .

٣ - الان جاء دور الايبروم لارسال الداتا الى الماستر (ان جميع البتات السابقة كان يتم ارسالها من الماستر باتجاه الخادم وكان الخادم يرد باشارة ACK) اما الان فان الخادم (الايبروم) هو الذي سيقوم بارسال الداتا باتجاه الماستر ، حيث يقوم الخادم بارسال البايت (BYTE3) الذي تم تحديد عنوانه في البداية ، بعد استقبال البايت يرد الماستر بعدم ارسال اشارة وهذا يدعى (NO ACK) أي عدم ارسال اشارة علم .. **السؤال الان لماذا يقوم الماستر بعدم ارسال الاشارة للخادم؟؟؟**

الجواب كالتالي: بما ان الخادم يقوم بعملية الارسال ، فهذه طريقة حتى يقوم الماستر باخبار الخادم باكمال عملية ارسال البايتات او بايقاف العملية .. وهكذا فان (NO ACK) تخبر الخادم بالتوقف عن ارسال البايتات.

٤ - تنتهي عملية قراءة الداتا بارسال اشارة نهاية الارسال (STOP) من قبل الماستر .

المثال الثالث

والان لنقم بعمل برنامج باستخدام الميكروسي يقوم بقراءة القيمة من الموقع رقم 530 داخل الايبروم 24C16 وحفظها في المتغير (X) وذلك باستخدام بروتوكول النقل I2C

كما قلنا نقوم بتحديد التردد المستخدم ١٠٠ كيلو هيرتز من خلال الميكروسي كالتالي:

```
I2C_Init(100000);
```

البايتات اللازمة لتحديد الموقع من خلال تسلسل عملية الكتابة يكون البرنامج الخاص بها كالتالي:

```

Char x;

Void main()

{

I2C1_Init(100000);

() I2C1_Start();

(I2C1_Wr(0xA4);

(I2C1_Wr(0x12);

}

```

لان يجب اعادة ارسال اشارة (START) مرة اخرى وهذا يستخدم امر جديد هو :

```
I2C1_Repeated_Start();
```

الان نكمل عملية القراءة .

البايت BYTE1 : هو نفس البايت الاول مع اختلاف البت الثامن ، فالبت الثامن يساوي ١ ، وهكذا يعرف الخادم بان العملية هي عملية قراءة :

Byte1'=1010 010 1=A5

البايت الثالث BYTE3 : هو البايت الذي سيتم قراءته من الايبروم وستم حفظه في المتغير (X) من خلال امر القراءة :

X=I2C_Read(0);

كما قلنا فان الماستر يرسل (NO ACK) بعد استقبال البايت وهذا يعني وضع القيمة صفر في الاقواس داخل امر القراءة هكذا يكون البرنامج كاملا كالتالي:

```

int X;
void main ()
{
I2C1_Init(100000);
I2C1_Start();
I2C1_Wr(0xA4);
I2C1_Wr(0x12);
I2C1_Repeated_Start(); I2C1_Wr(0xA5);
X=I2C1_Rd(0);
I2C1_Stop();
}

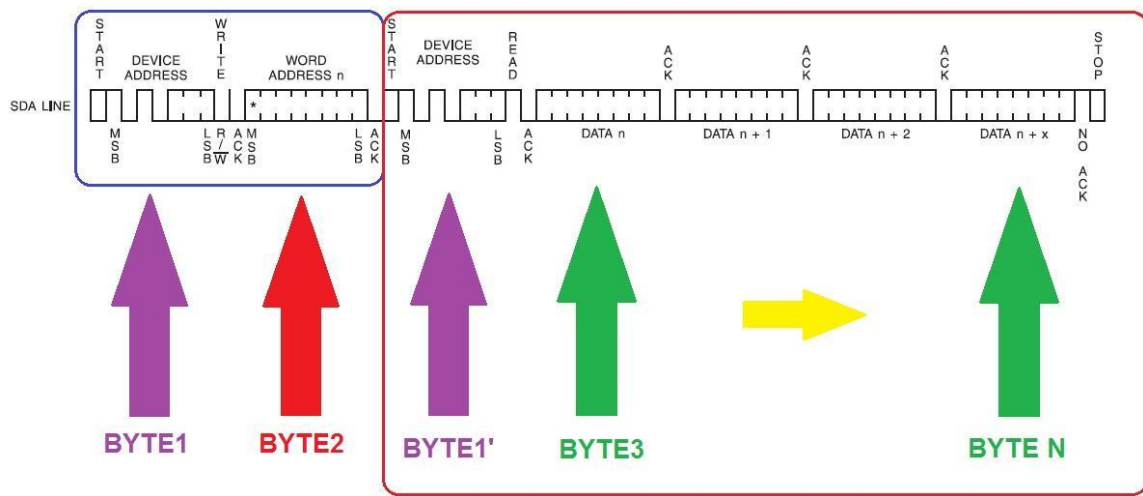
```

في البرنامج السابق تمت قراءة الموقع 530 في الايبروم ثم حفظه في المتغير x .. يمكن استخدام القيمة المقروءة في أي عملية نحتاجها . فمثلا يمكن قراءتها واظهارها على الاطراف الخارجية وغيرها من الوظائف .

ننتقل الى النوع الثاني وهو :

٢ - قراءة عدة بايتات متتالية من الايبروم (Page Read) .

لا تختلف عملية قراءة عدة بايتات متتالية عن طريقة بايت واحد كثيرا .. والصورة التالية توضح ذلك :



تتكون عملية القراءة من قسمين ، القسم الايسر المحدد باللون الازرق تم شرحه سابقا من خلال عملية الكتابة ، ووظيفته وضع مؤشر الموقع في المكان الذي سيتم القراءة منه كالتالي:

١ - ارسال اشارة بدء الارسال (START) من قبل الماستر .

٢ - يقوم الماستر بارسال البايث الاول BYTE1 والذي يتكون من 7 بتات لتحديد العنوان المميز للخادم ، اما البت الثامن فيساوي الصفر ، وهكذا يفهم الخادم بان العملية هي عملية كتابة الى الايبروم ، ، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال قد تمت بنجاح .

٣ - يقوم الماستر بارسال البايث الثاني BYTE2 والذي يحوي العنوان الذي سيتم الكتابة اليه داخل الايبروم (هذا العنوان يتم وضعه في مؤشر الموقع للايبروم) ، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال قد تمت بنجاح

نلاحظ ان هذه الخطوات هي نفس الخطوات المستخدمة للكتابة الى الايبروم ولكن مع الغاء اخر خطوة ، الان اصبح مؤشر الموقع في المكان الذي نريد التعامل معه .

الان نبدأ عملية القراءة من الموقع بعد ان قمنا بتحديد مكان مؤشر الموقع ، وهذا محدد في القسم الايمن المحدد باللون الاحمر كالتالي:

١ - يقوم الان الماستر بارسال اشارة بدء ارسال (START) مرة اخرى .

٢ - يقوم الماستر بارسال البايت ('BYTE1) وهو نفس البايت الاول (BYTE1) الذي يحدد العنوان المميز للخادم مع اختلاف واحد ، فالبت الثامن اصبح يحتوي القيمة (١) مما يعني ان العملية هي قراءة من الايبروم، بعدها يرد الخادم باشارة ACK والتي تعني بان عملية الاستقبال للعنوان تمت بنجاح .

٣ - الان جاء دور الايبروم لارسال الداتا الى الماستر (ان جميع البتات السابقة كان يتم ارسالها من الماستر باتجاه الخادم وكان الخادم يرد باشارة ACK) اما الان فان الخادم (الايبروم) هو الذي سيقوم بارسال الداتا باتجاه الماستر ، حيث يقوم الخادم بارسال البايت (BYTE3) الذي تم تحديد عنوانه في البداية ، بعد استقبال البايت الاول يرد الماستر باشارة علم ACK والتي تعني للخام ان يستمر بارسال البايت في الموقع التالي ، وهكذا يستمر الخادم في ارسال البايتات المتتالية حتى يستقبل اشارة (NO ACK) عندها يتوقف الخادم عن ارسال البايتات .

٤ - تنتهي عملية قراءة الداتا بارسال اشارة نهاية الارسال (STOP) من قبل الماستر .

نلاحظ ان العملية هي نفسها مع اختلاف بسيط وهو انه بدلا من ارسال اشارة NO ACK بعد قراءة البايت الاول فان الماستر يرسل ACK باستمرار حتى يحصل على اخر بايت عندها يرسل NO ACK ليوقف الخادم عن ارسال البايت التالي

المثال الرابع

والان لنقم بعمل برنامج باستخدام الميكروسي يقوم بقراءة 3 قيم ابتداء من الموقع رقم 680 داخل الايبروم 24C16 وحفظها في المتغيرات (X1,X2,X3) وذلك باستخدام بروتوكول النقل I2C .

هكذا يكون البرنامج كاملا كالتالي:

```
int x1,x2,x3;
void main()
{
I2C1_Init(100000);
I2C1_Start();
I2C1_Wr(0xA4);
I2C1_Wr(0xA8);
I2C1_Repeated_Start();I2C1_Wr(0xA5);
x1=I2C1_Rd(1);
x2=I2C1_Rd(1);
x3=I2C1_Rd(0);
I2C1_Stop();
}
```

انتهت المحاضرة الحادية عشر